

ORIGINAL PAGE IS  
OF POOR QUALITY

## Systolic Arrays and Stack Decoding

M. Shahshahani

Communications Systems Research Section

*This article reviews the work of K. Yao and C. Y. Chang on the application of systolic priority queues to the sequential stack decoding algorithm. Using a systolic array architecture, one can significantly improve the performance of such algorithms at high signal-to-noise ratio. However, their applicability at low SNR is doubtful.*

### I. Introduction

An active area of current research on deep space communications is the development of codes usable at low signal-to-noise ratio. The requirements for such codes are low error probability and the practicality of the decoding algorithm. It is well known that the error probability of convolutional codes decreases with their constraint lengths. However, the complexity of the Viterbi algorithm, which is the standard method for decoding convolutional codes at low SNR, has exponential growth with the code's constraint length. In a report on research conducted under a JPL contract, Yao and Chang suggest that, by using a systolic array architecture, decoding procedures for long constraint length codes can be practically implemented. The viability of the sequential stack algorithm (using systolic arrays) as an alternative to Viterbi's method is the main contention of these authors. This article reports on the scope and limitations of their approach. A serious limitation of their approach is that it may not be useful at the low signal-to-noise ratio for deep space communication.

### II. Stack Algorithm

The encoding procedure for a convolutional code can be regarded as a route through the code tree in the usual manner. A received symbol sequence is then a path in the code

tree. To every path,  $x$ , is associated a real number  $m(x)$  called the Fano metric. Figure 1 is a schematic description of the stack algorithm for decoding (see [1] for further details).

The Fano metric is constructed with the maximum likelihood criterion in mind. In fact, it is a generally accepted theorem that the stack algorithm is a good approximation to maximum likelihood decoding at high SNR.

The simplicity of the stack algorithm, as compared to Viterbi's, is reflected in the design of the hardware for the decoder. The wiring problem for the Viterbi decoder becomes extremely complicated for convolutional codes of constraint length  $>15$ , while the same problem remains fairly simple for the stack algorithm. Exponential growth of the layout area with the constraint length is another serious problem for VLSI design of large constraint length Viterbi decoders. For the stack algorithm, the growth of the layout area is only linear with the constraint length.

In spite of these advantages, the stack algorithm has remained unpopular for several reasons. Most notable are:

- (1) The reordering of paths according to the metric requires a large memory and is very time consuming. This often leads to overflow of the buffer and erasures.

- (2) The performance of the stack algorithm at low SNR is considerably inferior to that of the Viterbi algorithm.

For low data rate and/or two-way communications, erasures (frame deletions) are not a serious problem. However, for application to deep space communication, this effect presents a major obstacle. This problem will be discussed in more detail in section III.

In their articles on the application of systolic priority queues to sequential decoding ([2] and private communication contained in a report "Systolic Array Processing for Stacking Algorithms" which was submitted to Jet Propulsion Laboratory, Pasadena, California as a Second Progress Report), Chang and Yao note that a complete reordering of paths is not necessary for the implementation of the stack algorithm. In fact, the choice of the best path, i.e., the path with the largest Fano metric, is the only requirement and this can be accomplished efficiently by an application of systolic priority queues. Unfortunately, no quantitative measure of the improvement in efficiency is provided by Chang and Yao.

Roughly speaking, systolic arrays, as introduced by Kung [3] and applied by Leiserson to priority queues [4], are a form of parallel processing that has found many applications in VLSI design. Several designs for systolic queues for the determination of the best path are available, viz., random access memory (RAM), shift register scheme (SRS), and ripple register scheme (RRS). A general problem faced by many parallel processing schemes is the necessity of insertion of global controls for proper synchronization of the system. Chang and Yao recommend the RRS since it does not require global controls and maintains the local communication property (private communication contained in a report "Systolic Array Processing for Stacking Algorithms" which was submitted to Jet Propulsion Laboratory, Pasadena, California as a Second Progress Report).

It is also noted by Chang and Yao [5] that the Viterbi algorithm can be regarded as a matrix-vector multiplication (here, matrix entries are from an algebra where multiplication is defined as taking minimum). Therefore, this algorithm lends itself to parallel processing, and systolic priority queues can be used for improvement of the Viterbi decoder. The idea of using parallel processing for VLSI design of the Viterbi decoder is, of course, not new, and substantial work has already been done in this area by researchers at JPL ([6], [7], and [8]).

### III. Performance Statistics

It is clear from the description of the stack algorithm that the number of computations necessary to advance one

node in the code tree is a random variable  $N$ . At low signal-to-noise ratio, one encounters situations where backtracking is necessary and this effectively increases the mean of the random variable  $N$ . The behavior of  $N$  has been studied by a number of workers in coding theory. In particular, Jacobs and Berlekamp [9] obtained a lower bound for  $N$ . They showed that for fixed error or erasure probability, the distribution of  $N$  satisfies the following bound:

$$P(N > t) > t^{-\alpha} (1 + o(t))$$

It is important to note that this bound depends only on the channel error probability and the code rate and is independent of the choice of the method for selection of the best path. The code rate  $R$  and the exponent  $\alpha$  are functionally dependent. As  $\alpha$  tends to 1 from below, the mean of  $N$  approaches infinity. The value  $R_0$  of  $R$  corresponding to  $\alpha = 1$  is called the cut-off rate. Sequential decoding for  $R > R_0$  is practically impossible, since the number of computations becomes exceedingly large.

In "A Simulation Study for the Stack Algorithm for Low SNR" (a preprint article submitted to Jet Propulsion Laboratory, Pasadena, California), Chang reports on his simulation results on stack decoding for a (24, 1/4) convolutional code when  $\text{SNR} = E_b/N_0$  is between 0.9 and 1.3 dB. As pointed out by the author himself,  $R = 1/4$  is greater than the cutoff rate. Therefore, for *a priori* reasons, one cannot draw any optimistic conclusions about the performance of stack decoding on the basis of this work. Moreover, a comparison of this data and those of S. Z. Kalson (JPL Internal Document, Memo 331-86.2-217, November 6, 1986), for a (15, 1/5) convolutional code, shows that the performance of the stack decoder at  $\text{SNR} = 0.9$  dB is comparable to that of the Viterbi decoder at  $\text{SNR} = 0.4$  dB. This comparison of the Viterbi and stack algorithms did not take into account the overhead due to the short frame length (= 100 bits) adopted by Chang. The loss due to the overhead for a marker of length  $\lambda$  and frame length  $L$  is  $10 \log(1 + \lambda/L)$ , where  $\log$  is taken to base 10. Thus, for a 32-bit marker, the loss is about 1.2 dB.

Some of the key parameters chosen by Chang for his study appear to be unrealistic. As pointed out earlier, the code rate 1/4 and the frame length 100 are hardly acceptable choices for these parameters. Chang also gives no indication of the nature of the (24, 1/4) code he is using for his simulation. Special attention must be paid in the choice of the code, since different codes of the same constraint length and rate perform differently under sequential decoding. A discussion of what constitutes a "good" code for sequential decoding appears in [10]. It is also clear that the buffer size affects the error probability and the performance of the stack algorithm. For a conclusive study of the possibilities of the stack algorithm at low SNR, the following points should be kept in mind:

- (1) Experiment with several, much longer ( $>1000$ ) frame lengths and lower rate codes. This would substantially reduce the overhead and clarify the dependence of the error probability (mainly frame deletion) on the frame length.
- (2) Make sure the chosen code is a "good" one.
- (3) Quantify the dependence of error probability on the buffer size and the computation time. (The latter point is addressed by Chang.)

- (4) The effect of using soft decision on the performance of the stack algorithm should be clarified.

#### IV. Conclusion

The application of systolic array architecture, as suggested by Chang and Yao, is a significant improvement in the sequential stack decoding techniques at high signal-to-noise ratio. However, it is unlikely that the stack algorithm can serve as a viable alternative to the Viterbi algorithm at low SNR.

### References

- [1] G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*. New York: Plenum Press, 1981.
- [2] C. Y. Chang and K. Yao, "Systolic Array Architecture for the Sequential Stack Algorithm," *SPIE Proc.*, vol. 696, pp. 196-203, August 1986.
- [3] H. T. Kung, "Let's Design Algorithms for VLSI Systems," *Proc. of Caltech Conference on VLSI*, Caltech Comp. Sci. Dept., pp. 65-90, 1979.
- [4] C. E. Leiserson, *Area Efficient VLSI Computation*. Cambridge, Massachusetts: MIT Press, 1983.
- [5] C. Y. Chang and K. Yao, "Systolic Array Processing of the Viterbi Algorithm," *Proc. 23rd Annual Allerton Conference on Comm., Control and Computing*, Sponsored by Co-ordinated Sci. Lab of the Dept. of Electrical Engineering and Comp. Engineering of the U. of Illinois at Urbana-Champaign, pp. 430-439, Oct. 1985, (First Progress Report submitted to JPL, 1986).
- [6] F. Pollara, "Viterbi Algorithm on the Hypercube," *Proc. 23rd Annual Allerton Conference on Comm., Control and Computing*, Sponsored by Co-ordinated Sci. Lab of the Dept. of Electrical Engineering and Comp. Engineering of the U. of Illinois at Urbana-Champaign, pp. 440-449, Oct. 1985.
- [7] F. Polara, "Concurrent Viterbi Algorithm with Traceback," *SPIE Proc.*, vol. 696, pp. 204-209, Aug. 1986.
- [8] I. S. Hsu, T. K. Truong, I. S. Reed, and J. Sun, "A New VLSI Architecture for Viterbi Decoders of Large Constraint Length Convolutional Codes" (preprint), Pacific Rim Conf. Victoria, Canada, IEEE, New York, June 1987.
- [9] I. M. Jacobs and E. R. Berlekamp, "A Lower Bound to the Distribution Computation for Sequential Decoding," *IEEE Trans. Inf. Theory*, IT-13, pp. 167-174, April 1967.
- [10] S. Lin and D. J. Costello, *Error Control Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

ORIGINAL PAGE IS  
OF POOR QUALITY

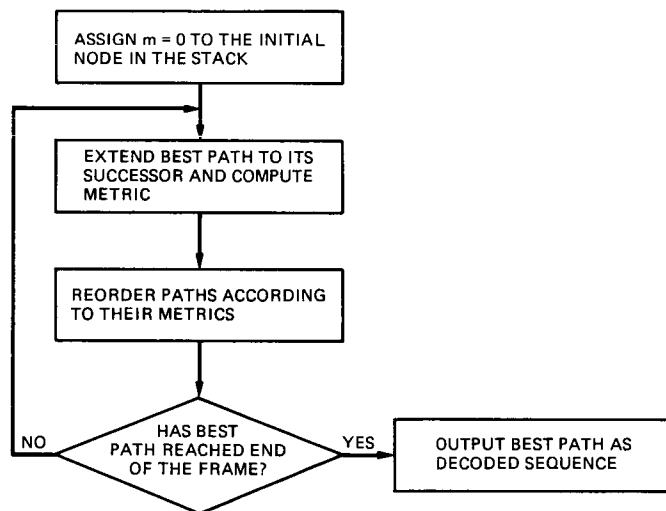


Fig. 1. Flow chart for stack decoding